CMPE 58N - Lecture 0.
Monte Carlo methods

Introduction, Course structure, Motivating Examples, Applications

Department of Computer Engineering,

Boğaziçi University, Istanbul, Turkey

Instructor: A. Taylan Cemgil

Goals of this Course

- Provide a basic understanding of underlying principles of Monte Carlo computation
- Orientation in the literature
- Focus on computational techniques rather than technical details,
  - ... the focus is not on proofs
  - ... but there will be some maths
    - Probability Theory
    - Statistics
    - Calculus and Linear Algebra
- Sharpening your intuition

Topics

- Markov Chain Monte Carlo
- Sequential Monte Carlo
- Probability theory
  - General background
  - Applications

Main study materials

- Handouts, Papers
- Jun S. Liu, Monte Carlo Strategies in Scientific Computing, 2001, Springer.
- Adam M. Johansen and Ludger Evers (edited by Nick Whiteley), Monte Carlo Methods, Lecture notes, University of Bristol

  http://www.maths.bris.ac.uk/~manpw/teaching/notes.pdf

- Information Theory, Inference, and Learning Algorithms
  David MacKay, Cambridge University Press – fourth printing (March 2005)

  http://www.inference.phy.cam.ac.uk/mackay/itprnn/book.html

General background about probability theory

- Geoffrey Grimmet and David Stirzaker, Probability and Random Processes, (3rd Ed), Oxford, 2006
  - Companion book containing 1000 exercises and solutions
- Grinstead and Snell, Introduction to probability available freely online!

  `http://www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/book.htm`

Main Book on Monte Carlo techniques

- Jun S. Liu,
  Monte Carlo Strategies for Scientific computing,
  Springer 2004
  - Short book
  - Covers almost everything we will mention on MCMC and SMC + more
  - Rather dense and Is not very easy to read

Other Books on Monte Carlo techniques

- Gilks, Richardson, Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman Hall, 1996
- Doucet, de Freitas, Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001

Tutorials and overviews (check course web page)

- Andrieu, de Freitas, Doucet, Jordan. *An Introduction to MCMC for Machine Learning*, 2001
- Andrieu. *Monte Carlo Methods for Absolute beginners*, 2004
- Doucet, Godsill, Andrieu. ”On Sequential Monte Carlo Sampling Methods for Bayesian Filtering”, Statistics and Computing, vol. 10, no. 3, pp. 197-208, 2000

Course Structure

- Web page

  `http://www.cmpe.boun.edu.tr/courses/cmpe58N/2009spring/`
- Required Work
  - Weekly Assignments (Reading, Programming, Analytic Derivations)
  - A project proposal and outline
  - Final Project: Implementation and Report
- Testing
  - 1 Midterm (in class), 1 Final (take home)
- Grading
  - Relative weights
    - % 25 Midterm
    - % 25 Take home final exam
    - % 50 Assignments, Quizzes and Final Project

Possible Topics

- In one application area (including but not limited to)
  - Scientific data analysis (DNA, Bioinformatics, Medicine, Seismology)
  - Robotics, Navigation, Self Localisation
  - Signal, Speech, Audio, Music Processing
  - Computer Vision (Object tracking)
  - Information Retrieval, Data mining, Text processing, Natural Language Processing
  - Sports, Finance, User Behaviour, Cognitive Science e.t.c.
- Reading a paper and writing a tutorial-like summary in own words and self designed examples
- Implementation and comparative study of inference algorithms on synthetic data

Remarks

- If you have already chosen a research topic
  - Use the project of this course to implement and write up a component of your work!
- If you have **not** chosen research/thesis topic but roughly have something in mind or simply don't know yet
  - Come and talk to me to clarify a topic/technique
  - Study/learn a few inference techniques more in depth
    - Never underestimate the insight gained from a well designed toy example
  - Investigate the feasibility/suitability of Monte Carlo techniques for your purposes

Remarks

- Ideally, a good report could be presented with some extensions at a national or international conference
  - Some well-known methods were master theses once,
  - Occasions when a forth year project report was published (and cited later!)
- Use TEX or LATEX.
  - If you are serious with research in computer science, statistics or engineering but are using other ways of document preparation, it is very likely that you are wasting some of your valuable time.

Remarks

- Any programming language or other system for computation and visualisation
  - Matlab (preferred)
  - Octave
  - Java,
  - C/C++, BLAS, ATLAS, GNU Scientific Library

Applications of Monte Carlo

- Statistics, Bioinformatics
- Signal Processing, Machine learning
- Seismology, Acoustics
- Computer Science, Analysis of algorithms, Randomized algorithms
- Networks, System simulation
- Robotics, Tracking, Navigation
- Econometrics, Finance
- Operations Research, Combinatorics, Optimisation
- Physics, Chemistry, Computational Geometry
- Environmental sciences, monitoring

Bayesian Statistics

- Computation of analytically intractable high dimensional integrals
- Inference, Model selection

Probabilistic Inference

- **expectations** of functions under probability distributions: **Integration**

$$\langle f(x) \rangle = \int_{\mathcal{X}} dx p(x) f(x) \qquad \langle f(x) \rangle = \sum_{x \in \mathcal{X}} p(x) f(x)$$

- **modes** of functions under probability distributions: **Optimization**

$$x^* = \operatorname*{argmax}_{x \in \mathcal{X}} p(x) f(x)$$

- However, computation of multidimensional integrals is hard

Combinatorics

- Counting

Example : What is the probability that a solitaire laid out with 52 cards comes out successfully given all permutations have equal probability ?

$$|A| = \sum_{x \in \mathcal{X}} [x \in A] \qquad [x \in A] \equiv \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

$$p(x \in A) = \frac{|A|}{|\mathcal{X}|} = \frac{?}{\approx 2^{225}}$$

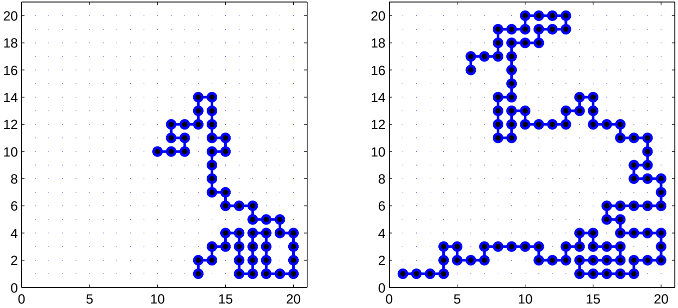
Random Combinatorial Objects

Generate uniformly from

- Self avoiding random walks on a $N \times N$ grid
- All spanning trees of a graph
- Binary trees with N nodes
- Directed Acyclic Graphs

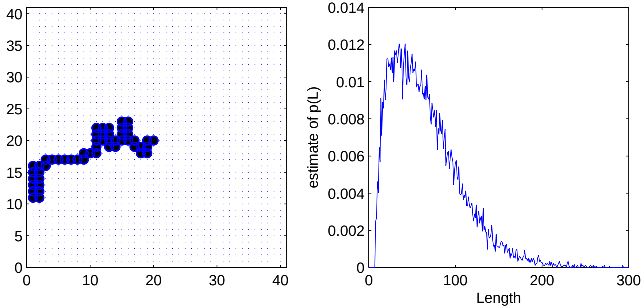
Self avoiding random walks

- How many different ways are there to place a chain with M nodes on an $N \times N$ 2-D rectangular grid ?
- In 3-D, problem is relevant for understanding protein folding



Self avoiding random walks

- S



Random Spanning Trees

- Given an undirected graph, generate a spanning tree **uniformly** from the set of all spanning trees
- (Aldous and Fill):

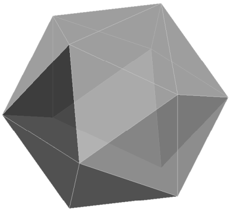
  Run a random walk on the graph until all verticies have been visited,
  Include the edge that the walk first visited v
  It turns out that the spanning tree generated like this is an uniform draw.

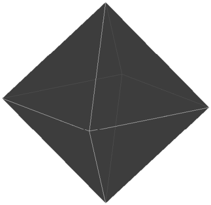
Geometry

- Given a simplex S in N dimensional space by

$$S = \{x : Ax \leq b, \quad x \in \mathbb{R}^N\}$$

find the Volume $|S|$





Sampling uniformly from a set S

- Suppose we have a black box implementation of an indicator function $[x \in S]$
- How can we generate uniform samples from S?
- It turns out that the following algorithm works (in principle)

  Choose an arbitrary $x^{(0)} \in S$
  For $i = 1, 2, \ldots$

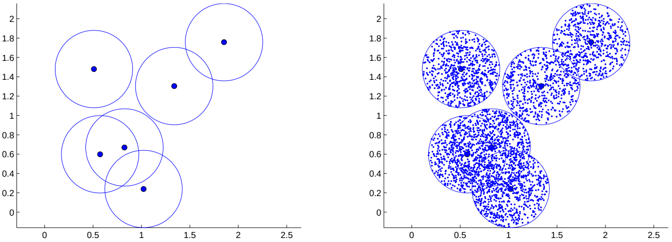
    Propose:
    $\epsilon_i \sim \mathcal{N}(0, V)$
    $x' \leftarrow x^{(i-1)} + \epsilon_i$
    Accept/Reject
    if $[x' \in S]$ then $x^{(i)} \leftarrow x'$ else $x^{(i)} \leftarrow x^{(i-1)}$ endif

  EndFor

  - $x^{(i)}$ are the desired samples! Why?

Sampling uniformly from a set S

$$S = \{x : \|c_i - x\| \leq \rho\}$$

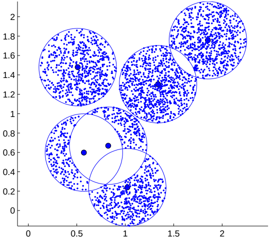


Sampling uniformly from a set S

```
x = c(:,1);
for i=1:5000,
   xhat = x + 0.2*randn(2,1);
   % Inclusion test
   e = c - repmat(xhat, [1 N]);
   d = sqrt(sum(e.^2,1));
   if any(d<rho),
      x = xhat;
      line(x(1), x(2), 'marker', '.');
   end;
end;
```

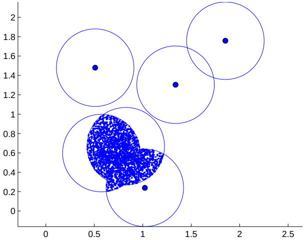
Sampling uniformly from a set S

Set of points that are close only a single center.
$S = \{x : \|c_i - x\| \leq \rho \text{and} \|c_j - x\| \geq \rho \text{for} i \neq j\}$



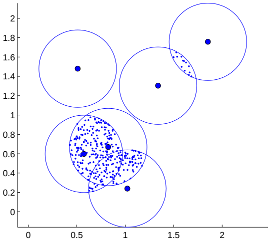
Sampling uniformly from a set S

Set of points that are close to two or more centers.



Sampling uniformly from a set S

```
%    xhat = x + 0.2*randn(2,1);
     xhat = x + 0.9*randn(2,1);
```



Matrix Permanent

- We define a so-called *restriction matrix* A where $A_{i,j} \in \{0, 1\}$.
- We think of A as an adjacency matrix of a bipartite graph $\mathcal{G}_A = (\mathcal{V}_s, \mathcal{V}_t, \mathcal{E})$
- $A_{i,j} = 1 \Leftrightarrow s_i \in \mathcal{V}_s, t_j \in \mathcal{V}_t, (i, j) \in \mathcal{E}$
- permanent(A) = total number of perfect matchings on $\mathcal{G}_A$
- (Vailant 1977) Problem is $\sharp P$ (harder than NP!). But Jerrum et.al. developed a polynomial time randomised algorithm based on simulating a Markov chain with known mixing time!

Network analysis, Rare Events

- Given a graph with random edge lengths

$$x_i \sim p(x_i)$$

Find the probability that the **shortest path from A to B** is larger than γ.